EE314 Digital Electronics Laboratory 2017-2018 Spring Term Project Final Report An FPGA Based Oscilloscope

1st Halil TEMURTAS *2094522* *halil.temurtas@metu.edu.tr*
2nd Erdem TUNA *2167419* *erdem.tuna@metu.edu.tr*

***Abstract*—The design of an FPGA based digital oscilloscope**

***Index Terms*—FGPA, oscilloscope, ADC, VGA**

I. Introduction

In this project, it is intended to realize a digital oscilloscope by using an FPGA. The FGPA will receive an analog signal, digitize it through an ADC; then will calculate the required parameters. Lastly data and input signal will be displayed on a VGA screen. The overall diagram is shown in *Figure 1*. The implementation logics and algorithms are discussed in respective sections.

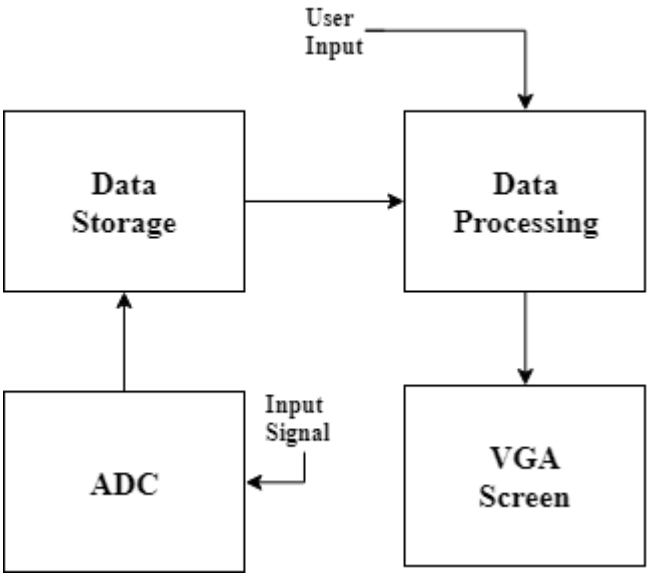


Figure 1: The Block Diagram of the Project

II. ALTERA DE1-SoC

In this project, we have used ALTERA DE1_SoC Development Kit for main unit and a VGA Monitor as a screen for the oscilloscope. In this part, our aim is to introduce the FPGA used in the project. The overall look of the device can be seen at *Figure 2*, as can be seen from the figure, the Development Kit consists of multiple parts aside from FPGA. In this project, GPIO Pins, seven segment display, push buttons, switches and VGA output of DE1-SoC was used.

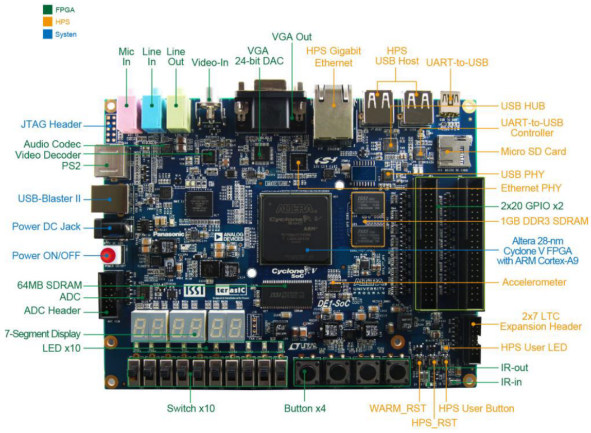


Figure 2: ALTERA DE1_SoC Development Kit

GPIO pins, also known as general-purpose input/output pins, were used for getting the analog input that is desired to be monitored through oscilloscope. As can be seen from the *Figure 3*, the basic circuitry for this pins includes Schottky diodes for protection, and a resistor.

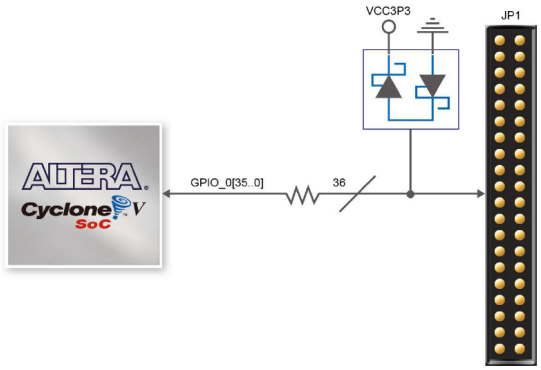


Figure 3: GPIO Pins

Seven segment display was used for testing the output values of computation value without using external monitor that we had some difficulties to find. As can be seen from the *Figure 4*, every led on the seven segment is connected through a resistor to the FPGA.

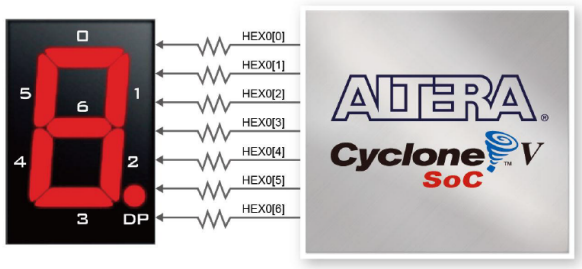


Figure 4: Seven Segment Display

Finally, the switch buttons were used as a mode buttons of the oscilloscope and push buttons were used as a division changer for the oscilloscope. The connections to FPGA can be seen at *Figure 5* and *Figure 6*. Finally the VGA connection used will be explained later in the report.

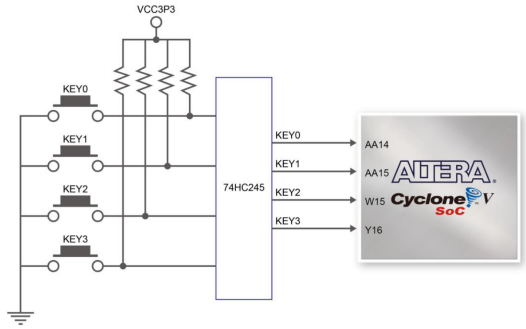


Figure 5: Push Buttons

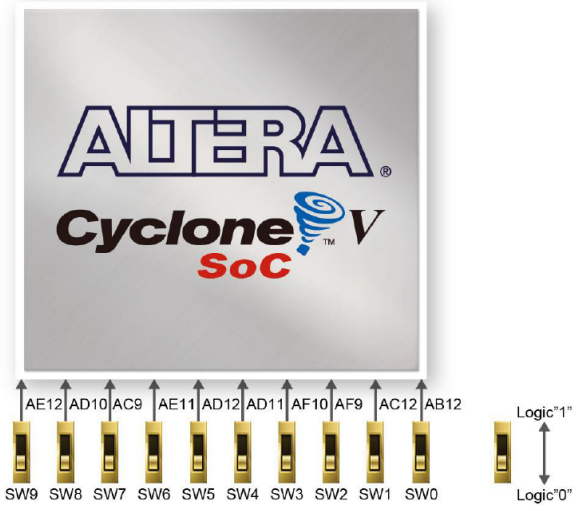


Figure 6: Switch Buttons

III. Modules

The modules and functional blocks designed and used in the project are presented in this section.

A. ADC

This module quantizes the analog input data. The hardware used is LTC2308 that is built into the FPGA. The Altera provides an example code regarding the use of the built-in ADC[1]. By evaluating the provided functionalities of this example with the help of a MsC project [2], a code is written in Verilog HDL and in Qsys environment. Qsys instantiates the internal connections of embedded modules to use ADC controller. The resulting Qsys module is exported as BSF file and that is shown in *Figure 7*. This will be utilized as an ADC block. The code is written according to I/O ports of this block. But the ultimate ADC bus is output from "conduit_end_SDI" port of the ADC block.

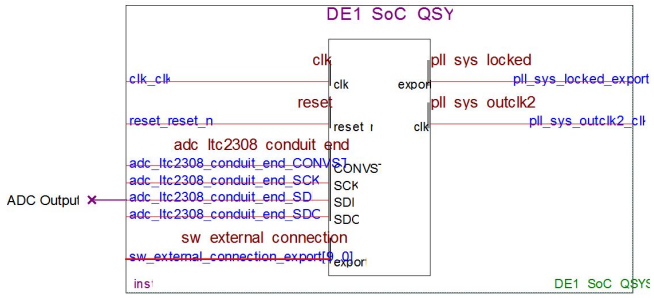


Figure 7: The Block Diagram of the ADC

B. RAM

The main function of RAM is to introduce a block to the system that can be both writable and readable. Writing is necessary to save data coming from ADC and reading is also necessary to process the data and show them on VGA screen. The RAM should work with FIFO structure to realize proper operation on VGA screen. "A FIFO (first-in-first-out) buffer is an "elastic" storage between two subsystems."[3]. A FIFO based read and write operation is depicted in *Figure 8*. The RAM is designed in behaviour level however, yet not implemented. Possible state diagram for the RAM Module can be seen at *Figure 9*.

C. Computation Unit

Computation unit is the unit responsible for all kinds of mathematical calculation of the project. For instance mean value calculation for the input signal are conducted in this unit. This Unit can be considered as a brain of the project. Some important parts are as follows,

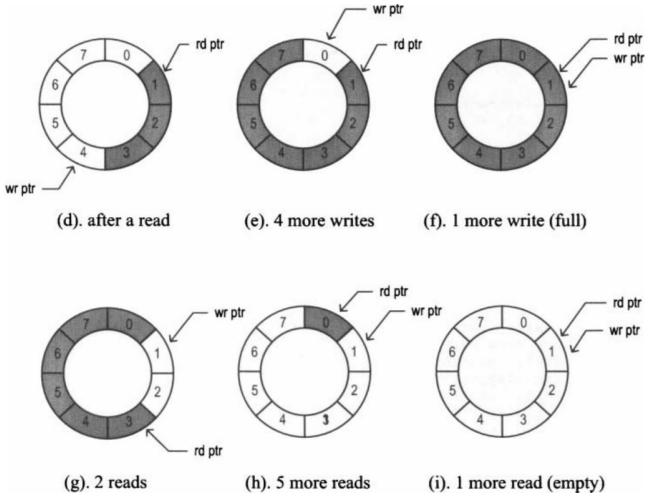


Figure 8: FIFO Working Principle[3]

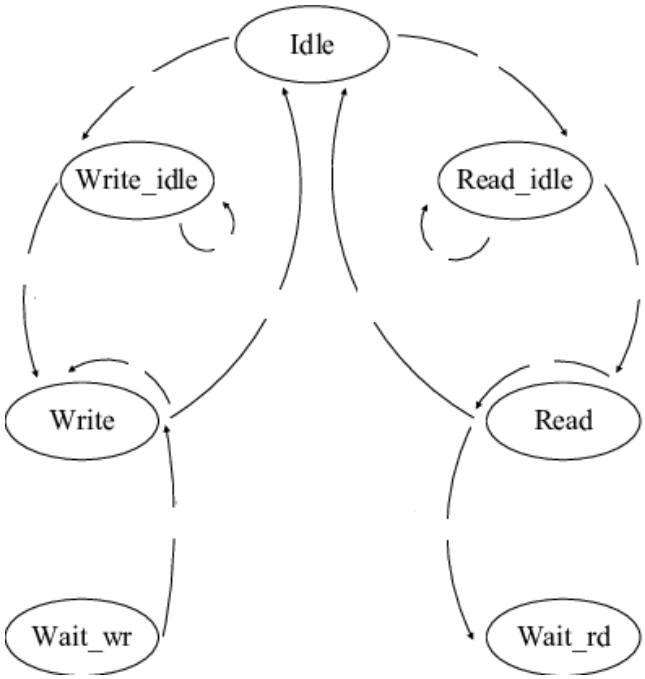


Figure 9: State Diagram for RAM

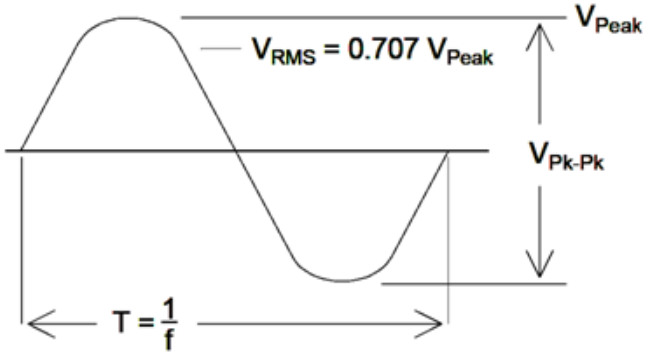


Figure 10: Voltage Waveform and its Properties

a) Peak-to-Peak Value Calculation

To find the peak-to-peak value of the input voltage, we needed to find the maximum and minimum voltage values of the given input. For that purpose we have used a well known sorting algorithm. Initially a point on the waveform was picked and every point next to it was compared to it. If the next point on the voltage is bigger than the initially stored value, the stored value is replaced with that value. After tracing every point on the waveform, the biggest value can be found. Similarly, the lowest value on the voltage waveform can be found.

After finding minimum and maximum points at the voltage waveform, peak-to-peak value is the difference between these values as can be seen from the *Figure 10*.

b) Mean Value Calculation

c) DC Offset Calculation

DC offset is a mean amplitude displacement from

d) Frequency Calculation

D. Mode Selection (AC/DC)

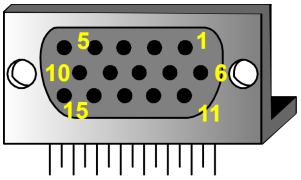
AC/DC Modes are one of the most fundamental mods of digital oscilloscopes in market. In this part we will build a module to build our own selection mode. One slide switch will be assigned to retrieve the desired mode information from the user. Based on this information, screen will display the input signal either with the DC offset or without the DC offset.

1) AC Mode: In AC Mode operation of the oscilloscope, the DC offset voltage is removed from the input voltage before it is reflected to the VGA monitor. For that Computation Unit will be used to extract offset information from stored data.

2) DC Mode: In DC Mode operation of the oscilloscope, the DC offset voltage is untouched from the stored data of the input voltage. The stored data is reflected directly to the VGA monitor.

E. VGA Screen

VGA is a widely used standard in video industry for the transmission of video signals from a computer or microprocessor into a monitor or TV. The input pin configuration for the VGA Monitor can be seen at *Figure 11*. VGA provides 640x480 pixel resolution. This resolution, however, is not the total pixel in horizontal and vertical axes. There is a blank border frame surrounding the display area. The horizontal axis has 48 pixels of border width on the left and 16 pixels of border width on the right side of the screen. Additionally, there are 96 pixels of border width for ray tube the retrace through the diagonal of the screen. The vertical axis has also similar additional pixels. 10 and 33 pixels for top and bottom borders, respectively. Also 2 pixels for the vertical retrace of the ray tube. The total structure of a VGA screen can be seen in *Figure 13*. The whole screen is 800x525 pixels.



Pin 1: Red	Pin 5: GND
Pin 2: Grn	Pin 6: Red GND
Pin 3: Blue	Pin 7: Grn GND
Pin 13: HS	Pin 8: Blu GND
Pin 14: VS	Pin 10: Sync GND

Figure 11: VGA Input Pins

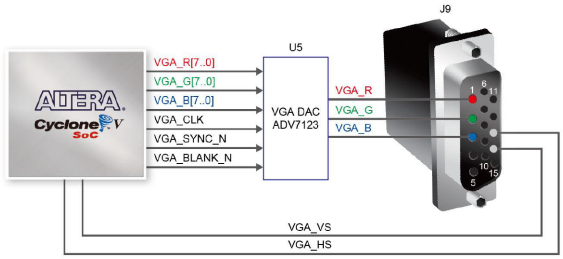


Figure 12: The Block Diagram of the Project

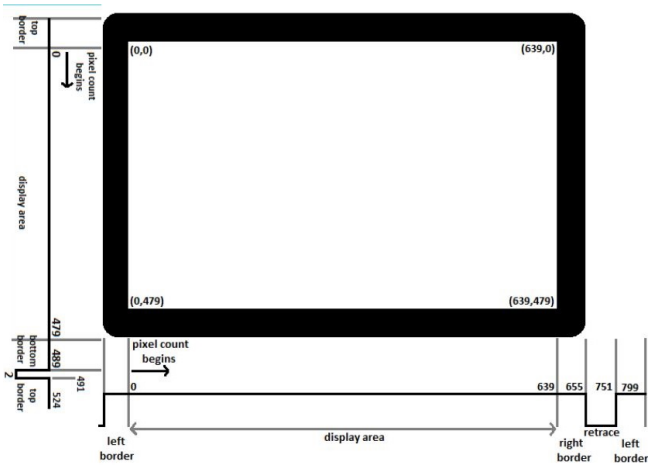


Figure 13: The Block Diagram of the Project

1) VGA Controller:

The VGA controller combines the data from RAM and Computation Unit to create a signal that can be displayed on the VGA monitor. Each of the RAM Modules contains an image that is ready for display on the screen. However, the data must be positioned relative to each other and combined. Also this module performs once a second as desired in the project requirements. The VGA controller also gets data from different data inputs such as Time/div and Voltage/div in order to reflect the waveform as user requires. In this part multiple clock signals and counters needed to display the waveform accordingly. Basic VGA Controller can be seen at *Figure 14*[4].

This module will supply an input data for the VGA controller for user preferences. Two push buttons will be assigned for the Time/div inputs that can be considered as Time+/div and Time-/div. As user pushes to Time+/div button, the time scale will be larger than the previous value. Similarly, as the user pushes to Time-/div button, the time scale will be smaller than the previous value. According to user preferences, this module allows user to see wider or narrower parts of input waveform.

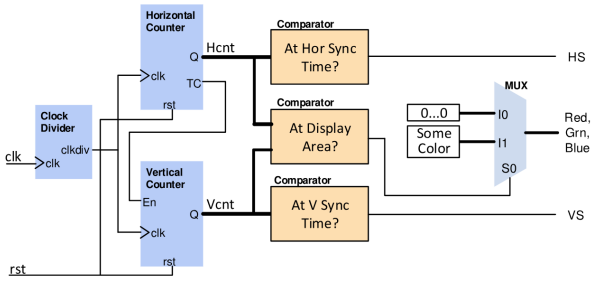


Figure 14: VGA Controller

2) Voltage/div Input:

This module will supply an input data for the VGA controller for user preferences. Two push buttons will be assigned for the Voltage/div inputs that can be considered as Voltage+/div and Voltage-/div. As user pushes to Voltage+/div button, the voltage scale will be larger than the previous value. Similarly, as the user pushes to Voltage-/div button, the voltage scale will be smaller than the previous value. According to user preferences, this module allows user to fit the voltage waveform to the screen.

3) Autoscale Input:

This module will also supply an input data for the VGA controller for user preferences. Since we are planning to use all push buttons for other modules. One slide switch will be assigned to this module. As the user triggers the button, this module will scale the waveform for the display such that it fits the display best.

IV. Conclusion

In this project, our aim is to design a FPGA based digital oscilloscope using Verilog. The project consists of five main part that are Analog to Digital Converter (VGA), RAM, Computation Unit, Mode Selection for Display and VGA Screen. Overall diagram of the project can be seen at *Figure 1*.

References

[1] "De1-Soc CD." [Online]. Available: http://download.terasic.com/downloads/cd-rom/de1-soc/DE1-SoC_v.5.0.1_HWrevF_SystemCD.zip. [Accessed: 21-May-2018].

[2] "Digital Scope Implemented on Altera DE1-SoC." [Online]. Available: https://people.ece.cornell.edu/land/courses/eceprojectsland/STUDENTPROJ/2015to2016/hj424/hj424_report_201605191237.pdf. [Accessed: 21-May-2018].

[3] P.P. Chu, Embedded SoPC Design with NIOS II Processor and Verilog Examples. New Jersey, 2012.

[4] "VGA Display Controller" [Online]. Avaible : https://learn.digilentinc.com/Documents/269